



CDW Documentation

NVIDIA GPU Firmware Upgrade Runbook (Revised)

NVIDIA GPU Firmware Upgrade Runbook

Process at a Glance

1. **Prepare** — record current versions, verify GPU health, stop all workloads
2. **Validate Package** — checksum the firmware file, preview what will change
3. **Flash GPU Tray** — update VBIOS, NVSwitch, EROT, FPGA via nvfwupd
4. **Reboot** — wait for background copy to complete, then cold reboot
5. **Flash Motherboard Tray** — update BMC, BIOS, PSUs; reset BMC
6. **Validate** — confirm new versions, GPU health, NVLink, smoke test

NOTE: Flashing the wrong firmware file or interrupting mid-flash can permanently brick a GPU. Read the full runbook before executing.

1. Overview

This runbook covers the end-to-end firmware upgrade process for NVIDIA data center GPUs. It is self-contained — do not consult external documentation mid-upgrade.

2. Background

What Is GPU Firmware?

Firmware is low-level software embedded in the GPU hardware itself. It controls boot behaviour, power management, and hardware communication — below the OS driver layer.

Components updated in a full DGX/HGX firmware upgrade:

Component	What It Controls
VBIOS	GPU boot, clocks, power limits, PCIe config
NVSwitch	GPU-to-GPU interconnect (NVLink fabric)
EROT	Hardware root of trust / secure boot
FPGA	Power management and control plane
BMC	Out-of-band server management
BIOS / UEFI	Motherboard initialisation
PSU	Power supply unit firmware

Why Upgrade?

- Fix GPU stability bugs or uncorrectable memory errors

- Apply security patches to the hardware trust chain
- Meet minimum firmware requirements for new driver versions
- Enable features such as Confidential Computing or NVLink enhancements

Minimum Driver Requirements

GPU Architecture	Minimum Driver
Ampere (A100, A30)	R450
Hopper (H100, H200)	R525
Ada Lovelace (L40, L40S)	R525

3. Tools

Tool Overview

Tool	Purpose	Where It Runs
nvfwupd	Full firmware bundle update for DGX/HGX via BMC Redfish API	Management host (remote)
NVFlash	VBIOS-only flash for standalone PCIe GPUs	GPU host (root required)
nvidia-smi	Query GPU health, firmware version, running processes	GPU host
nvsm	DGX system-wide health check	GPU host
ipmitool	Remote power cycle and BMC reset	Management host

Tool Selection by Platform

Platform	Tool
DGX H100 / H200	nvfwupd
HGX H100 / H200	nvfwupd
DGX A100	Firmware Update Container
Standalone PCIe GPU (A100, RTX etc)	NVFlash

Key Commands Reference

```
# --- nvfwupd ---
# Show installed vs package versions
nvfwupd -t ip=<BMC_IP> user=<USER> password=<PASS> show_version -p
<pkg.fwpkg>

# Update all firmware components
nvfwupd -t ip=<BMC_IP> user=<USER> password=<PASS> update_fw -p <pkg.fwpkg>
-y

# Update specific components only (with selector JSON)
nvfwupd -t ip=<BMC_IP> user=<USER> password=<PASS> update_fw -p <pkg.fwpkg>
-y -s selector.json

# Force downgrade controls
```

```

nvfwupd -t ip=<BMC_IP> user=<USER> password=<PASS> force_update enable
nvfwupd -t ip=<BMC_IP> user=<USER> password=<PASS> force_update status
nvfwupd -t ip=<BMC_IP> user=<USER> password=<PASS> force_update disable

# --- NVFlash (standalone PCIe GPUs only) ---
sudo nvflash --list # List all GPUs
and indices
sudo nvflash --version # Show VBIOS
version (GPU 0)
sudo nvflash --index=0 --save backup_$(date +%Y%m%d).rom # Backup current
VBIOS
sudo nvflash --index=0 new_firmware.rom # Flash new VBIOS
to GPU 0

# --- nvidia-smi ---
nvidia-smi -L # List all GPUs
nvidia-smi -q # Full GPU info
nvidia-smi -q | grep "VBIOS Version" # VBIOS version
only
nvidia-smi --query-compute-apps=pid,name,used_memory --format=csv # Active
GPU processes
nvidia-smi nvlink -s -i 0 # NVLink status
for GPU 0

# --- ipmitool ---
ipmitool -I lanplus -H <BMC_IP> -U <USER> -P <PASS> chassis power cycle #
Cold reboot
ipmitool -I lanplus -H <BMC_IP> -U <USER> -P <PASS> mc reset cold #
BMC-only reset
ipmitool -I lanplus -H <BMC_IP> -U <USER> -P <PASS> mc info #
Confirm BMC online
ipmitool -I lanplus -H <BMC_IP> -U <USER> -P <PASS> sel list #
BMC event log

# --- nvsm ---
nvsm show health # System-wide
health (DGX only)

```

4. Prerequisites

Complete every item below before starting. Do not proceed if any item cannot be confirmed.

Access & Credentials

- [] BMC IP address, username, and password (administrator privileges)
- [] SSH access to the target server
- [] Access to NVIDIA Enterprise Support Portal to download firmware packages

Firmware Package

- Correct .fwpkg file downloaded for your specific GPU model
- SHA256 checksum verified against NVIDIA release notes
- Package stored on the machine that will run nvfwupd

Environment

- All GPU workloads stopped — training jobs, inference servers, Kubernetes GPU pods
- `nvidia-smi` is not running — even passive queries block VBIOS updates
- Maintenance window scheduled and all stakeholders notified (minimum 2 hours)
- VBIOS backed up using NVFlash (standalone PCIe GPUs only)

System State

- Management host can ping the BMC IP
- Server is on stable power — no UPS-only or unreliable supply
- No other firmware operations are running on the same server
- Server reports healthy — confirmed via `nvsm show health` or BMC dashboard

Rollback Readiness

- Current firmware versions recorded in your change ticket
- Previous firmware .fwpkg file saved or accessible
- Rollback procedure reviewed (Section 9)

5. Best Practices

#	Practice	Why It Matters
1	Always run <code>show_version</code> first	Confirms exactly what will change before any flashing begins
2	Back up VBIOS before flashing	No backup = no rollback path for standalone GPU upgrades
3	Stop all GPU processes including <code>nvidia-smi</code>	Active processes block or corrupt the VBIOS update
4	Never interrupt mid-flash	Closing terminal or losing power mid-flash bricks the GPU
5	Verify the package checksum	A corrupted .fwpkg silently fails or causes a partial flash
6	Wait for <code>BackgroundCopyStatus: Completed</code>	Rebooting early means the firmware never actually activates
7	Use a wired management network	VPN or WiFi drops mid-update leave the flash incomplete
8	Disable <code>ForceUpdate</code> immediately after downgrade	Leaving it enabled is a security risk

#	Practice	Why It Matters
9	Validate before restoring workloads	Confirms the upgrade succeeded before returning node to use
10	Document pre and post versions in change ticket	Required for change management and future rollback reference

6. Phase 1 — Pre-Upgrade Preparation

Step 1.1 — Record Current Firmware Version

```
# DGX / HGX systems
nvwupd -t ip=<BMC_IP> user=<USER> password=<PASS> show_version

# Standalone PCIe GPU
nvidia-smi -q | grep "VBIOS Version"
```

Copy the full output to your change ticket. Fill in the table:

Component	Pre-Upgrade Version	Target Version
VBIOS		
NVSwitch FW		
EROT FW		
FPGA FW		
BMC		
System BIOS		

Step 1.2 — Verify GPU Health

```
nvidia-smi -L      # All expected GPUs must appear
nvsm show health  # All components must show OK (DGX only)
```

If any GPU is missing or any component is degraded — **stop**. Investigate before proceeding.

Step 1.3 — Stop All GPU Workloads

Stop all of the following before continuing:

- AI/ML training jobs, inference servers, GPU-accelerated applications
- Any monitoring script or agent that runs `nvidia-smi` on a schedule
- Kubernetes GPU pods — drain the node first:

```
kubectl drain <node-name> --ignore-daemonsets --delete-emptydir-data
```

Confirm no processes remain:

```
nvidia-smi --query-compute-apps=pid,name,used_memory --format=csv  
# Output must be completely empty before continuing
```

Step 1.4 – Backup VBIOS (Standalone PCIe GPUs Only)

Skip this step for DGX/HGX systems managed by nvfwupd.

```
sudo nvflash --index=0 --save gpu0_backup_$(date +%Y%m%d).rom
```

Store the .rom file on a separate host. This is your only rollback option if the flash fails.

Step 1.5 – Verify Package Integrity

```
sha256sum nvfw_DGX-HGX-H100-H200x8_<version>.fwpkg
```

Compare the output against the checksum in the NVIDIA release notes. If they do not match — re-download the package before proceeding.

Step 1.6 – Preview What Will Change

```
nvfwupd -t ip=<BMC_IP> user=<USER> password=<PASS> show_version -p  
<pkg.fwpkg>
```

Every component with a version mismatch will be updated. Confirm this matches your intent before proceeding.

7. Phase 2 – Firmware Upgrade Execution

Do not close your terminal, disconnect from the BMC, or allow the server to lose power once flashing begins. Any interruption mid-flash can permanently damage hardware.

7.1 DGX / HGX – nvfwupd

Step 2.1 – Flash the GPU Tray

Updates VBIOS, NVSwitch, EROT, and FPGA firmware for all GPUs in the system.

```
nvfwupd -t ip=<BMC_IP> user=<USER> password=<PASS> \  
update_fw -p nvfw_DGX-HGX-H100-H200x8_<version>.fwpkg -y
```

To update specific components only, create a selector JSON and pass it with -s:

```
{ "Components": ["VBIOS"] }
```

```
nvfwupd -t ip=<BMC_IP> user=<USER> password=<PASS> \  
update_fw -p <pkg.fwpkg> -y -s selector.json
```

Step 2.2 — Wait for Background Copy to Complete

The firmware is written to the GPU's flash storage in the background. **You must wait for this before rebooting.**

```
curl -k -u <USER>:<PASS> \  
https://<BMC_IP>/redfish/v1/Systems/HGX_Baseboard_0/Oem/Nvidia/BackgroundCopyStatus
```

Poll every 60 seconds. Only proceed when you see:

```
"BackgroundCopyStatus": "Completed"
```

If you see Failed — do **not** reboot. See the Troubleshooting section.

Step 2.3 — Cold Reboot the Server

```
ipmitool -I lanplus -H <BMC_IP> -U <USER> -P <PASS> chassis power cycle
```

Wait 5-10 minutes for the server to fully boot.

Step 2.4 — Flash the Motherboard Tray

After the server is back online, update BMC, BIOS, and PSU firmware:

```
nvfwupd -t ip=<BMC_IP> user=<USER> password=<PASS> \  
update_fw -p nvfw_DGX_<version>.fwpkg -y -s update_bmc.json
```

The motherboard tray uses a different .fwpkg file than the GPU tray. Check the NVIDIA release notes for the correct package name for your system.

Step 2.5 — Reset the BMC

```
ipmitool -I lanplus -H <BMC_IP> -U <USER> -P <PASS> mc reset cold
```

Wait 2-3 minutes, then confirm the BMC is back online:

```
ipmitool -I lanplus -H <BMC_IP> -U <USER> -P <PASS> mc info
```

7.2 Standalone PCIe GPU – NVFlash

Step 2.6 – Unload the NVIDIA Driver

```
sudo rmmod nvidia_uvm
sudo rmmod nvidia_drm
sudo rmmod nvidia_modeset
sudo rmmod nvidia

# Confirm all modules are unloaded (output must be empty)
lsmod | grep nvidia
```

Step 2.7 – Flash the VBIOS

```
# Flash GPU at index 0 – increment --index for additional GPUs (1, 2, ...)
sudo nvflash --index=0 new_vbios_firmware.rom
# NVFlash will prompt – review current vs new version shown, then type "yes"
```

Step 2.8 – Reboot

```
sudo reboot
```

8. Phase 3 – Validation

Step 3.1 – Confirm New Firmware Version

```
# DGX / HGX
nvwfupd -t ip=<BMC_IP> user=<USER> password=<PASS> show_version

# Standalone PCIe
nvidia-smi -q | grep "VBIOS Version"
```

Every updated component must now show the new version.

Step 3.2 – Verify GPU Detection

```
nvidia-smi -L
```

All expected GPUs must appear with correct model names. No Unknown or error entries.

Step 3.3 – Run Health Check

```
nvidia-smi -q          # Check per-GPU: VBIOS version, ECC mode, temperature,
power draw
nvsml show health     # DGX only – all components must show OK
```

Field	Expected
VBIOS Version	Matches new package version
ECC Mode	Enabled
Temperature	< 50°C at idle
Power Draw	Within TDP spec

Step 3.4 – Check NVLink (DGX / HGX Only)

```
nvidia-smi nvlink -s -i 0
```

All NVLink ports must show **Active**. Any inactive port must be investigated before restoring workloads.

Step 3.5 – Smoke Test

```
python3 -c "import torch; print(torch.cuda.is_available());
print(torch.cuda.get_device_name(0))"
```

Expected: True and the correct GPU model name.

Step 3.6 – Record Final State

Component	Pre-Upgrade	Post-Upgrade	Status
VBIOS			Pass / Fail
NVSwitch FW			Pass / Fail
EROT FW			Pass / Fail
FPGA FW			Pass / Fail
BMC			Pass / Fail
System BIOS			Pass / Fail

Step 3.7 – Restore Workloads

Only proceed after all checks above pass:

```
kubectl uncordon <node-name>
```

- Restart inference services and training jobs
- Re-enable GPU monitoring scripts

Monitor the system for 30–60 minutes after restoring workloads.

9. Rollback / Downgrade

Only downgrade if the new firmware causes a confirmed regression. Raise an NVIDIA Support case before attempting.

9.1 Standalone PCIe GPUs (NVFlash)

```
# Unload driver
sudo rmmod nvidia_uvm
sudo rmmod nvidia_drm
sudo rmmod nvidia_modeset
sudo rmmod nvidia

# Flash the saved backup
sudo nvflash --index=0 gpu0_backup_<date>.rom

# Reboot
sudo reboot
```

Then re-run all Phase 3 validation steps.

9.2 DGX / HGX Systems (nvfwupd)

```
# Step 1 – Enable ForceUpdate flag (required to allow downgrade)
nvfwupd -t ip=<BMC_IP> user=<USER> password=<PASS> force_update enable

# Step 2 – Confirm flag is active
nvfwupd -t ip=<BMC_IP> user=<USER> password=<PASS> force_update status

# Step 3 – Flash the previous firmware version
nvfwupd -t ip=<BMC_IP> user=<USER> password=<PASS> \
  update_fw -p nvfw_DGX-HGX-H100-H200x8_<old_version>.fwpkg -y

# Step 4 – Wait for BackgroundCopyStatus: Completed, then cold reboot (same
as Steps 2.2–2.3)

# Step 5 – MANDATORY: Disable ForceUpdate immediately after
nvfwupd -t ip=<BMC_IP> user=<USER> password=<PASS> force_update disable

# Step 6 – Confirm it is disabled
nvfwupd -t ip=<BMC_IP> user=<USER> password=<PASS> force_update status
```

Then re-run all Phase 3 validation steps.

10. Troubleshooting

Symptom	Cause	Action
"Invalid firmware file" error	Wrong .fwpkg for the component	GPU tray = HGX package; Motherboard tray = DGX package. Re-run <code>show_version</code> to confirm
VBIOS update blocked — GPU activity	Process still using the GPU	Run <code>nvidia-smi -query-compute-apps</code> and kill all PIDs. Drain K8s node. Retry update
BackgroundCopyStatus: Failed	Wrong firmware file or transient BMC error	Do not reboot. Retry the update command. Contact NVIDIA Support if retry fails
Server does not boot after reboot	Failed or incomplete VBIOS flash	Connect to BMC serial console. Check <code>ipmitool sel list</code> . Contact NVIDIA Support — do not power cycle again
GPU missing after upgrade	PCIe or driver issue	Check PCIe with <code>lspci</code> . Check <code>dmesg</code> for Xid errors. Reload driver. Run <code>nvsm show health</code>
NVLink ports show Inactive	NVSwitch FW issue or fabric manager down	Cold reboot. Restart <code>nvidia-fabricmanager</code> service. Contact NVIDIA Support if still inactive
Firmware version unchanged after reboot	Rebooted before background copy finished	Re-run the update. Wait for BackgroundCopyStatus: Completed. Reboot again
"Downgrade not allowed"	ForceUpdate flag not enabled	Follow Section 9.2 — enable ForceUpdate before running the downgrade command

11. Risk Register

Risk	Likelihood	Impact	Mitigation
Wrong firmware file flashed	Medium	Critical — GPU permanently bricked	Run <code>show_version</code> first; verify filename matches GPU model; check checksum
Power loss during flash	Low	Critical — GPU permanently bricked	Stable power only; never upgrade during power maintenance work
Workload active during upgrade	Medium	High — update fails or GPU degraded	Verify <code>nvidia-smi -query-compute-apps</code> output is empty before starting
Driver / firmware version mismatch	Medium	High — GPU fails to initialise	Check minimum driver requirements table before upgrading
Network loss to BMC mid-update	Low	High — incomplete flash	Use wired network; avoid VPN or WiFi for BMC management traffic
Previous firmware package unavailable	Medium	High — no rollback path	Save previous .fwpkg before every upgrade
ForceUpdate flag left enabled	Low	Medium — security risk	Always disable and verify immediately after any downgrade
Reboot before background copy completes	Medium	High — firmware does not activate	Always poll BackgroundCopyStatus until Completed before rebooting

12. References

- [NVIDIA DGX H100/H200 Firmware Update Guide](#) — primary reference: nvfwupd commands, update sequence, downgrade, troubleshooting
 - [NVIDIA DGX A100 Firmware Update Container Release Notes](#)
 - [NVIDIA Enterprise Support Portal](#) — firmware package downloads (login required)
-